**CODE REVIEW**

Кодот на Теодора Атанасовска според мене нема замерки и не е наппишан на многу комплициран начин, а и коментарите го олеснуваат разбирањето на истиот. Тој решава пробеми со проток на три начини (со Ford-Fulkerson, Edmond-Karps I Dinic алгоритмите), а исто така има и min-cut пресек.

Кодот се состои од две главни класи, GraphAdjLIst и GraphAdjMatrix, во истите се имплементирани истите функции, само во првиот со листи на соседство (матрица од вектор во вектор), а во другиот исто, но со матрици на соседство.

Првин ги иницијализира вредностите како што се колку темиња ќе чува, листа на соседство, капацитет на графот и капацитет на графот пред измени кои го чува за покасно првиот да во врати во почетна состојба. Потоа со addEdge функцијата ги додава ребрата на графот едно по едно преку main функцијата.

Edmond-Karps го изминува графот со bfs функцијата. Проверува од крајното ребро од аугментирачкиот пат па се до почетокот (со помош на parent низата) и го наоѓа минималниот flow. На секое ребро од аугментирачкиот пат му одзема од flow и додава flow на повратното ребро. Ова го прави се додека има аугментирачки пат од почеток до крај и секој пат пронајдениот flow го додава во променлива max\_flow за на крај да ја врати неа.Ford-Fulkerson го прави истото, ноединствената разлика е што користи dfs. Dinic ги користи и dfs и bfs со тоа што тој графот го дели на левели и потоа се движи по левелите исклучиво по растечки редослед користејќи bfs да ги направи левелите, а dfs да ги измине. После секое измиување со dfs го собира протокот за на крај да го даде максималниот проток. Има и min-cut функција која гледа кои темиња се достижни, а кои не.

Во main функцијата имплементираме два графа( едниот со листа на соседи, а другиот со матрица на соседи) и потоа им наоѓаме min-cut и максимален проток користејќи dinic. Кодот за истите примери работи како што треба.

Немам замерки ниту нешто што би сменил во кодот.

Марко Димитриевски